

PATENT APPLICATION
Docket No.: 13768.604.22

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of)
	Andy Harjanto)
)
Serial No.:	10/620,095)
) Art Unit
Conf No.:	8726) 2161
)
Filed:	July 15, 2003)
)
For:	METHOD AND SYSTEM FOR ACCESSING)
	DATABASE OBJECTS IN POLYARCHICAL)
	RELATIONSHIPS USING DATA PATH EXPRESSIONS)
)
Examiner:	Brent S. Stace)
)
Customer No.:	47973)
)

DECLARATION UNDER 37 C.F.R. § 1.131

DECLARATION UNDER 37 C.F.R. § 1.131

I, Andy Harjanto, declare as follows:

1. I am the inventor of the invention claimed in the above-identified patent application.
I am employed by Microsoft Corporation, the assignee of the above-identified patent application.
2. During my employment at Microsoft Corporation, and prior to January 2, 2003, I conceived and reduced to practice in the United States the Invention as described and claimed in the above-identified patent application.

3. Attached as Exhibit A is a copy of an email thread dated May 22, 2002. (hereinafter referred to as "Exhibit A"). Exhibit A includes my description of the invention including a number of examples.

4. I declare further that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful, false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful, false statements may jeopardize the validity of the application or any patent issuing thereon.

Dated this 18 day of December 2007.

Inventor:

Andy Harjanto



EXHIBIT A

From: Kim Cameron
Sent: Wednesday, May 22, 2002 9:58 AM
To: Andy Harjanto
Subject: RE: XPath, XQuery and Polyarchy
Importance: High

Very nice

-----Original Message-----

From: Andy Harjanto
Sent: Wednesday, May 22, 2002 1:20 AM
To: Kim Cameron; Jeff Parham
Cc: Chuck Chan; Stuart Kwan
Subject: XPath, XQuery and Polyarchy

After couple fun days of XQuery/ XPath prototyping, I began to realize the Polyarchy can be expressed very naturally into XPath/XQuery. This will enable many interesting queries that we would not even dare to think if they were in LDAP Filter form, or even SQL.

Note that the examples below are in Logical view (how the data viewed by the users), as opposed to how the data should be mapped to the store (Physical View).

Here is my proposal:

Consider the following XPath format:

```
\ -> XPath Root
  ViewName -> Immediate children (1st Level)
              RegularXPathExpression ----> 2nd, 3rd, 4th, etc...
```

The *ViewName*, extensible by the user, includes configuration information that describes:

- How the hierarchy data are related (e.g manager -> directReport).
- Where the logical root should start (e.g you do not have to start from BillG, you can start from DaveTh) The user should not be able to view beyond the logical root. It gives added security and performance benefits.
- ViewName can also be ACLed.
- Other future variables.

Simple XPath /Polyarchy Examples:

```
/Enterprise/Business/Platform/Windows/AD/andyhar
/Employee/BillG/SteveB/JimAll/BrianV/BillV/DaveTh/ChuckC
/ServerGroupEmployee/DaveTh/ChuckC
/AutoGroup/BrianVGroupFT/DaveFT/ChuckEmployees/andyhar
/Projects/Microsoft/Windows/Server/DirectoryServices/ActiveDirectory
```

/Office/US/Redmond/40/5234

/Printer/US/SEA/40/5/COPRXXA

/Identity/guid-2CC7C7B2-9B2D-11d3-9099-00A0C9E71419

/MyUDDI/MyBusiness/MyServices/2CC7C7B2-9B2D-11d3-9099-00A0C9E71439/5FD7C7B2-9A2D-31d3-9099-18A0C9E71439

Complex XPath/Poliarchy Examples:

All employees under DaveTh
 /ServerGroupEmployee/DaveTh/*

All employees under DaveTh work in Directory Project
 /ServerGroupEmployee/DaveTh/*[@ProjectName='Directory Service']

Recursively, get me all security enabled groups this user (who is in Chuck's Organization) belongs to
 /ServerGroupEmployee/DaveTh/ChuckC//[@LastName='Harjanto']/ancestor-self::*
 [@securityEnable=true]/@Name

Find me all US color printers that starts with CORP
 /Printer/US/*[@color=1]

Send all users in Building 40 a broadcast message
 /Office/US/Redmond/40/*/@userName

Combining with XQuery Examples:

Find all users who are under DaveTh and work in Building 40

```
FOR $x in document("ad")/ServerGroupEmployee/DaveTh/*
  $y in document("ad")/Office/US/Redmond/40/*
WHERE $x/@username = $y/@username
RETURN ...
```

Return all DaveTh's Direct Report in his/her reports in a hierarchical format

```
FOR $x in document("ad")/ServerGroupEmployee/DaveTh/*
RETURN
<DaveOrgChart>
  $x
</DaveOrgChart>
```

Return Union of all persons who worked in UDDI projects and developers who are at costCenter 12504

```
FOR $x in document("ad")/Projects/Windows/DirectoryServices/UDDI/*
```

```

    $y in document("ad")/Enterprise//*[costCenter='12504']
RETURN
  <Result>
    <$x/@Name>
    <$y/@Name>
  </Result>

```

Who Bob and Alice have in common, as far as manager?

```

FOR $x in document("ad")/OrgChart//*[@Name='Bob']/ancestor-or-self::*
  $y in document("ad")/OrgChart//*[@Name='Alice']/ancestor-or-self::*
WHERE $x = $y
RETURN
  <Result>
    <$x/@Name>
  </Result>

```

---- end ----

Wow, congrats, you've read until the end of examples -I owe you a cappuccino --- BTW: If you're interested in getting the demos, I think I could prototype some of these in a week...☺